Theoretical Computer Science 447 (2012) 53–61

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Robust optimization of graph partitioning involving interval uncertainty

Neng Fan [a,*], Qipeng P. Zheng [b], Panos M. Pardalos [a]

[a] *Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA*

[b] *Department of Industrial and Management Systems Engineering, West Virginia University, Morgantown, WV, USA*

ARTICLE INFO

Keywords:
Graph partitioning
Robust optimization
Uncertainty
Bipartite graph partitioning
Benders decomposition

ABSTRACT

The graph partitioning problem consists of partitioning the vertex set of a graph into several disjoint subsets so that the sum of weights of the edges between the disjoint subsets is minimized. In this paper, robust optimization models with two decomposition algorithms are introduced to solve the graph partitioning problem with interval uncertain weights of edges. The bipartite graph partitioning problem with edge uncertainty is also presented. Throughout this paper, we make no assumption regarding the probability of the uncertain weights.

1. Introduction

The graph partitioning problem is an NP-complete combinatorial optimization problem [7], and it consists of partitioning the vertex set of a graph into several disjoint subsets so that the sum of weights of the edges between the disjoint subsets is minimized.

Let $G = (V, E)$ be an undirected graph with a set of vertices $V = \{v_1, v_2, \dots, v_N\}$ and a set of edges $E = \{(v_i, v_j) :$ edge between vertices v_i and v_j, $1 \leq i, j \leq N\}$, where N is the number of vertices. The weights of the edges are given by a matrix $W = (w_{ij})_{N\times N}$, where w_{ij} (>0) denotes the weight of edge (v_i, v_j) and $w_{ij} = 0$ if no edge (v_i, v_j) exists between vertices v_i and v_j. This matrix is symmetric for undirected graphs G and is the adjacency matrix of G if $w_{ij} \in \{0, 1\}$.

Assume K is the number of subsets that we want to partition V into, and $C_{\min}$, $C_{\max}$ are lower and upper bounds of the cardinality of each subset, respectively. Usually, K is chosen from $\{2, \dots, N-1\}$ and $C_{\min}$, $C_{\max}$ can be chosen roughly from $\{1, \dots, N\}$ such that $C_{\min} \leq C_{\max}$.

Let x_{ik} be the indicator such that vertex v_i belongs to the kth subset if $x_{ik} = 1$ or not if $x_{ik} = 0$, and y_{ij} be the indicator such that the edge (v_i, v_j) with vertices v_i, v_j are in different subsets if $y_{ij} = 1$ and v_i, v_j in the same subset if $y_{ij} = 0$. Thus, the objective function of graph partitioning to minimize the sum of weights of edges connecting disjoint subsets can be expressed as $\min \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}y_{ij}$ or $\min \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij}$ because of $w_{ij} = w_{ji}$ and $w_{ii} = 0$ for non-existence of loops. Each vertex v_i has to be partitioned into one and only one subset, i.e., $\sum_{k=1}^{K} x_{ik} = 1$, and the kth subset has the number of vertices in range $[C_{\min}, C_{\max}]$, i.e., $C_{\min} \leq \sum_{i=1}^{N} x_{ik} \leq C_{\max}$. The relation between x_{ik} and y_{ij} can be expressed as $y_{ij} = 1 - \sum_{k=1}^{K} x_{ik}x_{jk}$ and this can be linearized as $-y_{ij} - x_{ik} + x_{jk} \leq 0$, $-y_{ij} + x_{ik} - x_{jk} \leq 0$ for $k = 1, \dots, K$ under the objective of minimization. Therefore, the feasible set of deterministic formulation of graph partitioning problem for a graph $G = (V, E)$ with weight matrix W is

* Corresponding author. Tel.: +1 352 214 5245.
E-mail address: andynfan@ufl.edu (N. Fan).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.10.015

$$X = \left\{ (x_{ik}, y_{ij}) : \begin{array}{l} \sum_{k=1}^{K} x_{ik} = 1, C_{\min} \le \sum_{i=1}^{N} x_{ik} \le C_{\max}, \\ -y_{ij} - x_{ik} + x_{jk} \le 0, \\ -y_{ij} + x_{ik} - x_{jk} \le 0, \\ x_{ik} \in \{0, 1\}, y_{ij} \in \{0, 1\}, \\ \quad i = 1, \ldots, N, j = i+1, \ldots, N, k = 1, \ldots, K \end{array} \right\}, \tag{1}$$

and the objective function is

$$\min_{(x_{ik}, y_{ij}) \in X} \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij}. \tag{2}$$

The *nominal graph partitioning problem* is to solve the program with the objective (2) and the constraints in (1) of X. This is a binary integer linear program.

The graph partitioning problem can be solved by the approaches of linear programming [11,4], quadratic programming [8,4] and semidefinite programming [14,9,11]. In this problem, the value of K can be given as a prior information or can be determined by the penalty method [4]. For the determination of $C_{\min}, C_{\max}$ for controlling the cardinalities of the disjoint subsets, we refer to discussions in [5]. Throughout this paper, we assume K and $C_{\min}, C_{\max}$ are given.

The previous optimization methods are all based on determinate data W and ignore the uncertainty. However, the weights of edges are not always constant and they are uncertain. For example, when analyzing the community structure in a social network [6], the relationship between two members is changing along time and it is uncertain. Therefore, the graph partitioning problem with uncertain weights of edges should be considered. There are two methods to address data uncertainty in mathematical programming models: stochastic programming and robust optimization. The stochastic programming method always requires the known probabilistic distributions of uncertain data, while robust optimization is to optimize against the worst cases by using a min–max objective [1]. Graph partitioning is a combinatorial optimization problem. In the past, robust version of many combinatorial problems have been studied, for example, the robust shortest path [12], the robust spanning tree [15] as well as many other problems in [10].

In this paper, we follow methods used in [1,2], which allow some violations and produce a feasible solution with high probability. The uncertainty we address in this paper is the interval uncertainty for weight matrix $W = (w_{ij})_{N \times N}$. Each entry w_{ij} is modeled as independent, symmetric and bounded random but unknown distribution variable $\tilde{w}_{ij}$ that takes values in $[w_{ij} - \hat{w}_{ij}, w_{ij} + \hat{w}_{ij}]$. Note that we require $w_{ij} = w_{ji}$ and thus $\hat{w}_{ij} = \hat{w}_{ji}$ for $i, j = 1, \ldots, N$. Assume $\hat{w}_{ij} \ge 0$, $w_{ij} \ge \hat{w}_{ij}$ and $w_{ii} = 0$ for all $i, j = 1, \ldots, N$.

In this paper, robust formulations for graph partitioning with uncertain W are discussed and several algorithms will be proposed based on the propositions of formulations. In addition, the cases for bipartite graph partitioning are also studied. This paper is also an extended version of our previous paper [6]. The rest of this paper is organized as follows: Section 2 discusses the formulations for the robust graph partitioning problem; in Section 3, two decomposition methods for solving the robust graph partitioning problem by solving a series of nominal graph partitioning problem are constructed; in Section 4, the bipartite graph partitioning problem involving uncertainty is discussed; Section 5 includes the computational results and analysis of these approaches; Section 6 concludes the paper.

2. Graph partitioning with uncertain weights

In this section, the robust optimization is to address the uncertainty of weight matrix W with $\tilde{w}_{ij} \in [w_{ij} - \hat{w}_{ij}, w_{ij} + \hat{w}_{ij}]$, where w_{ij} is the nominal value of edge (v_i, v_j). Let J be the index set of W with uncertain changes, i.e., $J = \{(i, j) : \hat{w}_{ij} > 0, i = 1, \ldots, N, j = i+1, \ldots, N\}$, where we assume that $j > i$ since W is symmetric. Let Γ be a parameter, not necessarily integer, that takes values in the interval $[0, |J|]$. This parameter Γ is introduced in [1,2] to adjust the robustness of the proposed method against the level of conservatism of the solution. The number of coefficients w_{ij} is allowed to change up to $\lfloor \Gamma \rfloor$ and another w_{i_t, j_t} changes by $(\Gamma - \lfloor \Gamma \rfloor)$. Thus formulation for the *robust graph partitioning problem* can be established as follows:

$$\min_{(x_{ik}, y_{ij}) \in X} \left(\sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} + \max_{\left\{ \substack{S : S \subseteq J, |S| \le \Gamma \\ (i_t, j_t) \in J \setminus S} \right\}} \left(\sum_{(i,j) \in S} \hat{w}_{ij} y_{ij} + (\Gamma - \lfloor \Gamma \rfloor) \hat{w}_{i_t, j_t} y_{i_t, j_t} \right) \right). \tag{3}$$

Since the value y_{ij} takes values from $\{0, 1\}$, $|y_{ij}|$ in the model [2] is reduced to y_{ij} here. Depending on the chosen of Γ, there are several cases: if $\Gamma = 0$, no changes are allowed and the problem reduces to nominal problem (2); if Γ is chosen as an integer, the maximizing part in (3) is $\max_{\{S|S \subseteq J, |S| \le \Gamma\}} \sum_{(i,j) \in S} \hat{w}_{ij} y_{ij}$; if $\Gamma = |J|$, the problem can solved by Soyster's method [13]. The index set J is equivalent to edge set E if all weights have uncertainty.

As shown in the following theorem, the problem (3) can be reformulated as an equivalent binary integer linear programming. The method used in this proof was first proposed in [2].

Theorem 1. *The formulation (3) is equivalent to the following linear programming formulation:*

$$\begin{aligned}
\min\ & \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \Gamma p_0 + \sum_{(i,j)\in J} p_{ij} \\
s.t.\ & p_0 + p_{ij} - \hat{w}_{ij}y_{ij} \geq 0, \quad \forall (i,j)\in J \\
& p_{ij} \geq 0, \quad \forall (i,j)\in J \\
& p_0 \geq 0, \\
& (x_{ik}, y_{ij}) \in X.
\end{aligned} \tag{4}$$

Proof. For given values $(y_{ij})_{i=1,\ldots,N,j=i+1,\ldots,N}$, the part

$$\max_{\left\{\substack{S:\, S\subseteq J,\, |S|\leq \Gamma \\ (i_t,j_t)\in J\setminus S}\right\}} \left(\sum_{(i,j)\in S} \hat{w}_{ij}y_{ij} + (\Gamma - \lfloor \Gamma \rfloor)\hat{w}_{i_t,j_t}y_{i_t,j_t} \right),$$

in (3) can be linearized by introducing z_{ij} for all $(i,j)\in J$ with the constraints $\sum_{(i,j)\in J} z_{ij} \leq \Gamma$, $0 \leq z_{ij} \leq 1$, or equivalently, by the following formulation

$$\begin{aligned}
\max\ & \sum_{(i,j)\in J} \hat{w}_{ij}y_{ij}z_{ij} \\
s.t.\ & \sum_{(i,j)\in J} z_{ij} \leq \Gamma, \\
& 0 \leq z_{ij} \leq 1, \quad \forall (i,j)\in J.
\end{aligned} \tag{5}$$

The optimal solution of this formulation should have $\lfloor \Gamma \rfloor$ variables $z_{ij} = 1$ and one $z_{ij} = \Gamma - \lfloor \Gamma \rfloor$, which is equivalent to the optimal solution in the maximizing part in (3).

By strong duality, for given values $(y_{ij})_{i=1,\ldots,N,j=i+1,\ldots,N}$, the problem (5) is linear and its duality can be formulated as

$$\begin{aligned}
\min\ & \Gamma p_0 + \sum_{(i,j)\in J} p_{ij} \\
s.t.\ & p_0 + p_{ij} - \hat{w}_{ij}y_{ij} \geq 0, \quad \forall (i,j)\in J \\
& p_{ij} \geq 0, \quad \forall (i,j)\in J \\
& p_0 \geq 0.
\end{aligned}$$

Combining this formulation with (3), we obtain the equivalent formulation (4), which finishes the proof. □

Our algorithm (**MIP**) is based on solving the binary linear program (4) directly by the CPLEX MIP solver [3]. There are $N \cdot K + \frac{N(N-1)}{2}$ binary decision variables with at most $\frac{N(N-1)}{2} + 1$ continuous variables in this formulation.

3. Decomposition methods for robust graph partitioning problem

3.1. Benders decomposition to a nominal problem and a linear program

In the formulation (4) for the robust graph partitioning problem, the variables p_0, p_{ij} are continuous while x_{ik}, y_{ij} are binary. For the fixed $\bar{x}_{ik}, \bar{y}_{ij}$, the formulation (4) can be reformulated as follows:

$$\begin{aligned}
\min\ & \Gamma p_0 + \sum_{(i,j)\in J} p_{ij} + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}\bar{y}_{ij} \\
s.t.\ & p_0 + p_{ij} \geq \hat{w}_{ij}\bar{y}_{ij}, \quad \forall (i,j)\in J \\
& p_{ij} \geq 0, \quad \forall (i,j)\in J \\
& p_0 \geq 0.
\end{aligned}$$

This is a linear program, and we can obtain its dual problem as follows:

$$\begin{aligned}
\max\ & \sum_{(i,j)\in J} \hat{w}_{ij}\bar{y}_{ij}z_{ij} + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}\bar{y}_{ij} \\
s.t.\ & \sum_{(i,j)\in J} z_{ij} \leq \Gamma, \\
& 0 \leq z_{ij} \leq 1, \quad (i,j)\in J.
\end{aligned} \tag{6}$$

By Benders decomposition method, the program (6) presents the subproblem. By solving this subproblem for giving values of $\bar{x}_{ik}, \bar{y}_{ij}$ at the iteration l, we can obtain the values $\bar{z}_{ij}^{(l)}$ and construct the optimality cut

$$z \geq \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} + \sum_{(i,j) \in J} \hat{w}_{ij} y_{ij} \bar{z}_{ij}^{(l)} \tag{7}$$

for the master problem. Therefore, the master problem for Benders decomposition method can be formulated as follows:

$$\begin{aligned} &\min \ z \\ &s.t. \quad z \geq \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} + \sum_{(i,j) \in J} \hat{w}_{ij} y_{ij} \bar{z}_{ij}^{(l)}, \quad l = 1, 2, \ldots, L \\ &\qquad (x_{ik}, y_{ij}) \in X. \end{aligned} \tag{8}$$

The program (6) is always feasible and bounded for any feasible solutions $\bar{x}_{ik}, \bar{y}_{ij}$ from (8), and thus, the feasibility cut can be eliminated in the master problem (8). Observing (6) and (8), the master problem is a binary integer linear program with respect to x_{ik}, y_{ij} and the subproblem is a linear program with respect to z_{ij}. Thus, by Benders decomposition method, we decompose the mixed integer program for the robust graph partitioning problem into a series of linear programs and mixed binary linear programs. Additionally, the subproblem (6) can be easily solved by a greedy algorithm: sorting the coefficients $\hat{w}_{ij}\bar{y}_{ij}$ of z_{ij} for $(i, j) \in J$ in the objective function in decreasing order; assigning the first $\lfloor \Gamma \rfloor$ corresponding z_{ij}s to be 1 according to this order; and assigning the last z_{ij} to be $\Gamma - \lfloor \Gamma \rfloor$ and all others $z_{ij} = 0$.

Theorem 2. *Solving the master problem* (8) *at iteration l is equivalent to solving l nominal graph partitioning problems.*

Proof. At the iteration l of Benders decomposition algorithm, there are l added optimality cuts in the form of (7), and this cut is equivalent to

$$z \geq \sum_{(i,j) \in J} (w_{ij} + \hat{w}_{ij} \bar{z}_{ij}^{(l)}) y_{ij} + \sum_{(i,j) \notin J} w_{ij} y_{ij}.$$

The right hand side of this cut is in fact the objective of a nominal graph partitioning problem with weights $w_{ij} + \hat{w}_{ij}\bar{z}_{ij}^{(l)}$ for edge $(i, j) \in J$ and w_{ij} for edge $(i, j) \notin J$. Therefore, solving the master problem at iteration l is equivalent to solving l nominal graph partitioning problems and then choosing the one with maximum objective. □

The algorithm (**BD**) based on Benders decomposition method is presented in the following table.

Algorithm **BD**

Step 1: Initialization:
$\quad\bar{x}_{ik}, \bar{y}_{ij} :=$ initial feasible solution in X for all i, j, k;
$\quad LB := -\infty, UB := \infty$;
Step 2: While there is gap larger than ε between UB and LB, i.e., $UB - LB > \varepsilon$, do the following steps:
$\quad$Step 2.1: Solve the subproblem (6) to obtain point $\bar{z}_{ij}$ for $(i, j) \in J$,
$\qquad$and add cut $z \geq \sum_{i=1}^{N} \sum_{j=i+1}^{N} w_{ij} y_{ij} + \sum_{(i,j)\in J} \hat{w}_{ij} y_{ij} \bar{z}_{ij}$
$\qquad$to the master problem (8);
$\qquad UB := \min\{UB, \sum_{j=i+1}^{N} w_{ij}\bar{y}_{ij} + \sum_{(i,j)\in J} \hat{w}_{ij}\bar{y}_{ij}\bar{z}_{ij}\}$;
$\quad$Step 2.2: Solve the master problem $\min\{z :$ added cuts, $x_{ik}, y_{ij} \in X\}$;
$\qquad LB := \bar{z}$, where $\bar{z}$ is the objective value of master problem;
Step 3: Output the optimal solution x_{ik}^*, y_{ij}^* for all i, j, k.

Step 1 of this algorithm requires finding a feasible solution. Here, we present a simple method for finding initial solutions of $\bar{x}_{ik}$s: putting vertices $v_1, v_2, \ldots, v_{C_{\min}}$ into the first subset, i.e., $\bar{x}_{11} = \bar{x}_{21} = \cdots = \bar{x}_{C_{\min},1} = 1$; putting the vertices $v_{C_{\min}+1}, v_{C_{\min}+2}, \ldots, v_{2C_{\min}}$ into the second subset, i.e., $\bar{x}_{C_{\min}+1,2} = \cdots = \bar{x}_{2C_{\min},2} = 1$; repeating these steps until we have $\bar{x}_{K\cdot C_{\min},K} = 1$; setting $\bar{x}_{(K\cdot C_{\min}+1),K} = 1, \bar{x}_{(K\cdot C_{\min}+2),K} = 1, \ldots, \bar{x}_{N,K} = 1$ and all other unassigned $\bar{x}_{ik}$s to be 0. The initial solution for $\bar{y}_{ij}$ can be obtained by $\bar{y}_{ij} = 1 - \sum_{k=1}^{K} \bar{x}_{ik}\bar{x}_{jk}$.

The Benders decomposition method can solve the robust graph partitioning problem by solving a series of nominal graph partitioning problems. However, for solving the master problem at iteration l, it is equivalent to solving l nominal graph partitioning problems. Although the Benders decomposition methods can converge in finite steps, say L, we need to solve $L(L+1)/2$ nominal graph partitioning problems totally. In next section, we present another decomposition method, which can take less computational time in some cases.

3.2. Algorithm based on the decomposition of one variable

For all $(i,j) \in J$, let e_l $(l = 1, \dots, |J|)$ be the corresponding value of $\hat{w}_{ij}$ in the increasing order. For example, $e_1 = \min_{(i,j)\in J} \hat{w}_{ij}$ and $e_{|J|} = \max_{(i,j)\in J} \hat{w}_{ij}$. Let $(i^l, j^l) \in J$ be the corresponding index of the lth minimum one, i.e., $\hat{w}_{(i^l,j^l)} = e_l$. In addition, we define $e_0 = 0$. Thus, $[e_0, e_1], [e_1, e_2], \dots, [e_{|J|}, \infty)$ is a decomposition of $[0, \infty)$.

For $l = 0, 1, \dots, |J|$, we define the program G^l as follows:

$$G^l = \Gamma e_l + \min_{(x_{ik},y_{ij})\in X} \left\{ \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq e_{l+1}} (\hat{w}_{ij} - e_l)y_{ij} \right\}. \tag{9}$$

Totally, there are $|J| + 1$ of G^ls. In the following theorem, we prove that the decomposition method based on p_0 can solve the program (4). The method in the proof was first proposed in [2].

Theorem 3. *Solving robust graph partitioning problem (3) is equivalent to solving $|J|+1$ problems G^ls in (9) for $l = 0, 1, \dots, |J|$.*

Proof. From (4) in Theorem 1, the optimal solution $(x^*_{ik}, y^*_{ij}, p^*_0, p^*_{ij})$ satisfies

$$p^*_{ij} = \max\{\hat{w}_{ij}y^*_{ij} - p^*_0, 0\},$$

and therefore, the objective function of (4) can be expressed as

$$\begin{aligned} &\min_{\{p_0\geq 0,(x_{ik},y_{ij})\in X\}} \Gamma p_0 + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j)\in J} \max\{\hat{w}_{ij}y_{ij} - p_0, 0\} \\ &= \min_{\{p_0\geq 0,(x_{ik},y_{ij})\in X\}} \Gamma p_0 + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j)\in J} y_{ij}\max\{\hat{w}_{ij} - p_0, 0\}, \end{aligned} \tag{10}$$

where the equality is obtained by the fact y_{ij} is binary in the feasible set X.

By the decomposition $[0, e_1], [e_1, e_2], \dots, [e_{|J|}, \infty)$ of $[0, \infty)$ for p_0, we have

$$\sum_{(i,j)\in J} y_{ij}\max\{\hat{w}_{ij} - p_0, 0\} = \begin{cases} \displaystyle\sum_{(i,j):\hat{w}_{ij}\geq e_l} (\hat{w}_{ij} - p_0)y_{ij}, & \text{if } p_0 \in [e_{l-1}, e_l],\ l = 1, \dots, |J|; \\ 0, & \text{if } p_0 \in [e_{|J|}, \infty). \end{cases}$$

Thus, the optimal objective value of (4) is $\min_{l=1,\dots,|J|,|J|+1}\{Z^l\}$, where

$$Z^l = \min_{\{p_0\in[e_{l-1},e_l],(x_{ik},y_{ij})\in X\}} \left(\Gamma p_0 + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq e_l} (\hat{w}_{ij} - p_0)y_{ij} \right), \tag{11}$$

for $l = 1, \dots, |J|$, and

$$Z^{|J|+1} = \min_{\{p_0\geq e_{|J|},(x_{ik},y_{ij})\in X\}} \Gamma p_0 + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij}.$$

For $l = 1, \dots, |J|$, since the objective function (11) is linear over the interval $p_0 \in [e_{l-1}, e_l]$, the optimal is either at the point $p_0 = e_{l-1}$ or $p_0 = e_l$. For $l = |J| + 1$, Z^l is obtained at the point $e_{|J|}$ since $\Gamma \geq 0$.

Thus, the optimal value $\min_{l=1,\dots,|J|,|J|+1}\{Z^l\}$ with respect to p_0 is obtained among the points $p_0 = e_l$ for $l = 0, 1, \dots, |J|$. Let G^l be the value at point $p_0 = e_l$ in (11), i.e.,

$$G^l = \Gamma e_l + \min_{(x_{ik},y_{ij})\in X} \left\{ \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij} + \sum_{(i,j):\hat{w}_{ij}\geq e_{l+1}} (\hat{w}_{ij} - e_l)y_{ij} \right\}.$$

We finish the proof. □

As shown in Theorem 3, $G^{|J|} = \Gamma e_{|J|} + \sum_{i=1}^{N}\sum_{j=i+1}^{N} w_{ij}y_{ij}$ is the original nominal problem with an added constant. Our Algorithm (**DP0**) is based on this theorem.

Algorithm **DP0**
Step 1: For all $(i,j) \in J$, sort $\hat{w}_{ij}$ in increasing order to obtain $e_0, e_1, \dots, e_{\lvert J\rvert}$;   Step 2: For $l = 0, 1, \dots, \lvert J\rvert$, solving G^l in (9);   Step 3: Let $l^* = \arg\min_{l=0,1,\dots,\lvert J\rvert} G^l$ and obtain the optimal solution $\{x^*_{ik}, y^*_{ij}\} = \{x_{ik}, y_{ij}\}^{l^*}$;

Algorithm (**DP0**) is based on the decomposition of $p_0 \in [0, \infty)$ and each subproblem G^l has the same computational complexity as the nominal graph partitioning problem. Since the nominal graph partitioning problem is NP-complete, from the decomposition algorithm (**DP0**), we can conclude that the robust graph partitioning problem is also NP-complete.

4. Bipartite graph partitioning involving uncertainty

The bipartite graph is defined as $G = (V, U, E)$ with vertex sets $V = \{v_1, \dots, v_N\}$, $U = \{u_1, \dots, u_M\}$ and edge set $E = \{(v_i, u_j) :$ edge between vertices v_i and u_j, $1 \le i \le N, 1 \le j \le M\}$, where N and M are the numbers of vertices within two sets, respectively. Usually, instead of weighted matrix, the biadjacency weighted matrix $A = (a_{ij})_{N\times M}$ is given where $a_{i,j}$ is the weight of edge (v_i, u_j). In [4,5], the relations for partitioning between graphs and bipartite graphs have been presented. Assume we still want to obtain K subsets of both V and U, and the cardinality for subsets of V is in the range $[C_{\min}, C_{\max}]$ and the cardinality for subsets of U is in the range $[c_{\min}, c_{\max}]$. Let the constraints of bipartite graph partitioning be a set as follows:

$$Y = \left\{ (x^v_{ik}, x^u_{jk}, y_{ij}) : \begin{array}{l} \sum_{k=1}^{K} x^v_{ik} = 1, C_{\min} \le \sum_{i=1}^{N} x^v_{ik} \le C_{\max}, \\ \sum_{k=1}^{K} x^u_{jk} = 1, c_{\min} \le \sum_{j=1}^{M} x^u_{jk} \le c_{\max}, \\ -y_{ij} \mp x^v_{ik} \pm x^u_{jk} \le 0, \\ x^v_{ik}, x^u_{jk}, y_{ij} \in \{0, 1\}, \\ \quad i = 1, \dots, N, j = 1, \dots, M, k = 1, \dots, K \end{array} \right\},$$

where $x^v_{ik}, x^u_{jk}, y_{ij}$ are the indicators for vertex sets V, U, and edge set E as the same explanations in Section 1.

The bipartite graph partitioning problem is formulated as

$$\min_{(x^v_{ik}, x^u_{jk}, y_{ij}) \in Y} \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij} y_{ij}.$$

Because of its similarity to the graph partitioning problem as discussed in Section 2, we also consider the uncertain of matrix $\tilde{A}$ where $\tilde{a}_{ij}$ takes values in $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$ for $i = 1, \dots, N, j = 1, \dots, M$. The robust optimization for uncertain $\tilde{a}_{ij}$ is as

$$\min_{(x^v_{ik}, x^u_{jk}, y_{ij}) \in Y} \quad \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij} y_{ij} + \max_{\left\{ \substack{S : S \subseteq J, |S| \le \Gamma \\ (i_t, j_t) \in J \setminus S} \right\}} \left(\sum_{(i,j) \in S} \hat{a}_{ij} y_{ij} + (\Gamma - \lfloor \Gamma \rfloor) \hat{a}_{i_t, j_t} y_{i_t, j_t} \right), \tag{12}$$

where $J = \{(i, j) : \hat{a}_{ij} > 0\}$ and $\Gamma \in [0, |J|]$.

As proved in Theorem 1, we can obtain the linear formulation as (4) for robust bipartite graph partitioning (12) similarly as follows:

$$\begin{aligned} \min \ & \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij} y_{ij} + \Gamma p_0 + \sum_{(i,j) \in J} p_{ij} \\ s.t. \ & p_0 + p_{ij} - \hat{a}_{ij} y_{ij} \ge 0, \quad (i, j) \in J \\ & p_{ij} \ge 0, \quad (i, j) \in J \\ & p_0 \ge 0, \\ & (x^v_{ik}, x^u_{jk}, y_{ij}) \in Y. \end{aligned} \tag{13}$$

We omit other methods and algorithms here since the robust optimization for bipartite graph partitioning is quite similar to graph partitioning problems.

5. Numerical experiments

In this section, all algorithms (**MIP, BD, DP0**) are implemented using CPLEX 11.0 [3] via ILOG Concert Technology 2.5, and all computations are performed on a SUN UltraSpace-III with a 900 MHz processor and 2.0 GB RAM. Computational times are reported in CPU seconds.

All tested graphs are randomly generated. The density r of a graph is the ratio of the number of edges and the number of possible edges. The uncertain values of $[w_{ij} - \hat{w}_{ij}, w_{ij} + \hat{w}_{ij}]$ are randomly generated. Here we assume $w_{ij} \in \{0, 1\}$ and

Table 1
Computational results and CPU seconds.

Graphs			Uncertainty		Objective values			CPU seconds		
N	r	K	$\|J\|$	Γ	MIP	BD	DP0	MIP	BD	DP0
10	0.1	3	4	2	0	0	0	0.01	0.01	0.01
	0.2		9	5	2.18	2.18	3.27	0.09	0.35	0.06
	0.3		13	7	1.03	1.03	1.67	0.10	0.23	0.05
	0.4		18	9	4.80	4.80	4.80	0.14	0.48	0.22
	0.5		22	11	7.68	7.68	8.59	0.39	0.65	0.20
	0.6		27	14	8.06	8.06	8.06	0.23	0.42	0.26
	0.7		30	15	11.58	11.58	11.58	0.73	1.21	0.37
	0.8		35	18	14.85	14.85	14.85	0.85	0.97	0.46
	0.9		40	20	18.08	18.03	18.29	0.89	4.13	0.28
15	0.1	3	10	5	0	0	0	0.01	0.03	0.01
	0.2		21	11	4.94	4.94	4.98	0.34	1.01	0.28
	0.3		31	16	4.55	4.55	4.55	0.75	0.56	0.39
	0.4		42	21	9.50	9.50	9.50	1.18	5.72	0.43
	0.5		52	26	11.86	11.78	11.86	2.16	2.20	0.60
20	0.1	3	19	10	0	0	0	0.01	0.04	0.03
	0.2		37	19	2.69	2.69	2.64	0.78	1.39	0.43
	0.3		57	29	6.58	6.58	6.58	1.11	1.56	0.70
	0.4		76	38	14.52	14.48	14.52	2.60	2.75	1.06
	0.5		95	48	15.65	15.65	16.25	3.25	11.59	1.48
20	0.1	4	19	10	0	0	0	0.01	0.06	0.02
	0.2		37	19	6.50	6.50	6.74	1.56	6.99	0.92
	0.3		57	29	12.01	12.01	13.11	3.32	9.96	1.54
	0.4		76	38	21.85	21.85	21.85	13.24	41.87	9.07
	0.5		95	48	25.65	25.05	25.65	11.05	21.04	4.74
30	0.1	3	42	21	1.04	1.04	1.46	0.86	0.51	0.07
	0.2		87	44	7.98	7.98	7.98	6.88	4.60	1.84
	0.3		130	65	10.64	10.64	10.64	4.34	7.43	2.39
30	0.1	4	42	21	2.19	2.19	2.29	2.40	6.09	1.30
	0.2		87	44	13.43	13.23	14.24	20.80	33.25	6.71
	0.3		130	65	16.44	16.44	16.44	12.60	20.47	7.49
40	0.1	3	78	39	3.41	3.41	3.41	5.07	3.79	1.07
	0.2		155	78	10.46	10.46	11.39	6.75	13.99	3.83
	0.3		233	117	20.64	20.64	22.00	58.36	134.24	35.57
40	0.1	4	78	39	5.46	5.46	6.89	3.83	11.83	2.49
	0.2		155	78	16.25	16.25	16.25	118.86	303.03	13.35
	0.3		233	117	31.94	31.61	32.23	229.26	>3000	259.06
50	0.1	3	122	61	1.73	1.73	1.73	2.02	1.46	0.12
	0.2		242	121	11.04	11.04	13.05	21.73	23.19	10.85
	0.3		366	183	25.65	25.65	25.65	2088.19	320.45	18.56
50	0.1	4	122	61	3.51	3.51	3.65	5.50	7.82	2.11
	0.2		242	121	19.64	19.94	21.65	1129.04	>3000	940.20
	0.3		366	183	39.71	39.71	42.63	>3000	>3000	>3000
50	0.1	5	122	61	5.43	5.43	5.43	10.80	20.76	4.65
	0.2		242	121	28.13	28.13	31.01	>3000	>3000	>3000

In this table, the the bounds for cardinalities are chosen as $1, N-1$, i.e., $C_{\min}=1, C_{\max}=N-1$.

$0<\hat{w}_{ij}/w_{ij}<1$ if $w_{ij}>0$. In Table 1, we assume the cardinality of each subset is in the range $[C_{\min}, C_{\max}]=[1, N-1]$. The gap in CPLEX is set to be 0.1. All objective values and computational seconds are presented in Table 1. From this table, we can find that the algorithm (**DP0**) is the most efficient one, and the algorithm (**BD**) is least efficient. As discussed in Sections 3.1 and 3.2, the algorithm (**BD**) needs to compute l nominal graph partitioning problems at iteration l, while the algorithm (**DP0**) computes $|J|+1$ nominal graph partitioning problems totally. Thus, if the Benders decomposition method cannot converges quickly in small number of iterations, it usually takes longer time than the algorithm (**DP0**).

Instead of loose cardinalities, we assume the bounds $C_{\min}, C_{\max}$ take values around N/K. All objective values and computational seconds for different graphs are presented in Table 2 with more conservative cardinality constraints. From this table, we can see in the case of same number of vertices, the computational times increase as the density increases.

Table 2
Computational results.

Graphs			Uncertainty		Cardinality	Objective values			CPU seconds		
N	r	K	$\|J\|$	Γ	$[C_{\min}, C_{\max}]$	MIP	BD	DP0	MIP	BD	DP0
10	0.1	3	4	2	[3,5]	0	0	0	0.01	0.02	0.02
	0.2		9	5		2.55	2.55	2.55	0.20	0.26	0.14
	0.3		13	7		6.29	6.21	6.03	0.17	0.61	0.24
	0.4		18	9		14.01	14.01	14.13	0.59	2.14	0.42
	0.5		22	11		18.97	18.97	19.19	0.80	13.16	0.52
	0.6		27	14		26.22	25.49	26.23	1.41	37.69	0.66
	0.7		30	15		27.65	27.59	28.80	1.93	42.98	0.85
	0.8		35	18		32.08	32.00	34.34	4.93	52.67	1.93
	0.9		40	20		38.52	37.82	39.63	5.01	241.77	3.54
15	0.1	3	10	5	[4,6]	1.31	1.31	1.69	0.07	0.12	0.07
	0.2		21	11		7.65	7.65	7.65	0.73	0.58	0.44
	0.3		31	16		18.67	17.92	18.06	0.87	4.45	0.45
	0.4		42	21		28.75	28.75	30.92	1.59	18.50	0.71
	0.5		52	26		40.58	40.49	40.95	10.26	479.57	1.78
20	0.1	3	19	10	[6,8]	5.35	5.29	5.62	0.41	0.41	0.28
	0.2		37	19		17.59	17.59	17.59	1.51	4.46	0.88
	0.3		57	29		34.27	34.03	34.22	3.54	42.34	2.10
	0.4		76	38		57.54	55.45	56.92	35.65	1358.55	16.83
	0.5		95	48		71.58	71.56	71.56	76.80	>3000	35.58
20	0.1	4	19	10	[4,6]	6.50	6.35	7.70	0.60	2.18	0.55
	0.2		37	19		22.03	21.78	22.10	13.79	115.28	2.98
	0.3		57	29		43.25	43.19	43.84	37.21	>3000	9.00
	0.4		76	38		65.37	65.27	65.27	209.51	>3000	46.65
	0.5		95	48		84.02	83.75	85.89	1190.84	>3000	346.60
30	0.1	3	42	21	[9,11]	13.87	13.87	15.12	1.72	5.26	1.21
	0.2		87	44		47.10	47.08	49.51	15.51	434.36	24.44
	0.3		130	65		86.16	84.56	86.66	358.65	>3000	113.79
30	0.1	4	42	21	[7,9]	17.59	17.58	17.59	5.71	85.38	2.67
	0.2		87	44		59.24	58.72	60.36	754.93	>3000	156.81
	0.3		130	65		106.62	104.48	105.34	>3000	>3000	2222.75

These results are based on same data sets as those in Table 1 and the same parameters except that $[C_{\min}, C_{\max}]$.

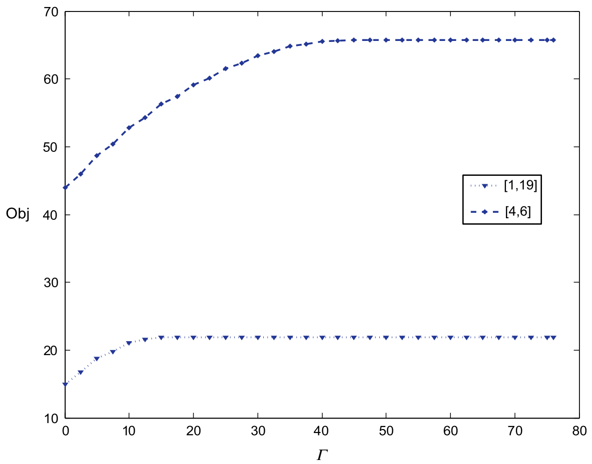

Fig. 1. Objective values regarding Γ. $N = 20$, $r = 0.4$, $|J| = 76$, $K = 4$, $[C_{\min}, C_{\max}] = [1, 19], [4, 6]$.

Comparing results in Tables 1 and 2, in the same graph, the case with loose cardinality constraints takes shorter time than the one with conservative bounds.

As discussed in Section 2, the parameter Γ is introduced in [1,2] to adjust the robustness of the proposed method against the level of conservatism of the solution. In Fig. 1, for the random generated graph with 20 vertices and density 0.4 with 76 uncertain edges, the relationship between the objective values and the values of Γ is presented for obtaining 4 subsets. From this figure, we can see that as the value of Γ increases for considering more uncertainties, the objective values are increasing as well. But when Γ is large enough, the objective value is a constant. Additionally, the values for the cases of conservative cardinalities are larger than corresponding cases of loose cardinalities.

6. Conclusions

In this paper, we present three algorithms for the robust graph partitioning problem with interval uncertain weights. We first present the formulation for this problem and then give the equivalent mixed integer linear programming formulation. Two decomposition methods, including Benders decomposition method and decomposition on one variable, can solve the robust graph partitioning problem by solving a series of nominal graph partitioning problems. We compare these algorithms on randomly generated graphs with uncertain weights. In this paper, a parameter Γ, introduced by [1,2], is chosen to allow some gap between the optimal value of the exact formulation and the robust solutions. Additionally, we study the bipartite graph partitioning problem involving uncertain weights.

The graph partitioning problem is NP-complete, and the robust graph partitioning problem is also NP-complete as shown in Section 3.2. Two decomposition algorithms, which are solving a series of nominal graph partitioning problems, can be used for further research. For example, the approximative semidefinite programming (SDP) method is useful for nominal graph partitioning problem, and we can combine these decomposition methods and the SDP method to solve large robust graph partitioning problems efficiently.

References

[1] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Math. Program, Ser. B 98 (2003) 49–71.
[2] D. Bertsimas, M. Sim, The price of robustness, INFORMS Oper. Res. 52 (1) (2004) 35–53.
[3] ILOG CPLEX 11.0 Users Manual, 2007.
[4] N. Fan, P.M. Pardalos, Linear and quadratic programming approaches for the general graph partitioning problem, J. Global Optim. 48 (1) (2010) 57–71.
[5] N. Fan, P.M. Pardalos, Multi-way clustering and biclustering by the Ratio cut and Normalized cut in graphs, J. Comb. Optim. (2010) doi:10.1007/s10878-010-9351-5.
[6] N. Fan, P.M. Pardalos, Robust optimization of graph partitioning and critical node detection in analyzing networks, in: W. Wu, O. Daescu (Eds.), COCOA 2010, Part I, in: LNCS, vol. 6508, 2010, pp. 170–183.
[7] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete graph problems, Theoret. Comput. Sci. 1 (1976) 237–267.
[8] W. Hager, Y. Krylyuk, Multiset graph partitioning, Math. Methods Oper. Res. 55 (2002) 1–10.
[9] S.E. Karisch, F. Rendl, Semidefinite programming and graph equipartition, in: P.M. Pardalos, H. Wolkowicz (Eds.), Topics in Semidefinite and Interior-Point Methods, American Mathematical Society, 1998, pp. 77–95.
[10] P. Kouvelis, G. Yu, Robust Discrete Optimization and its Applications, Kluwer Academic Publishers, 1996.
[11] A. Lisser, F. Rendl, Graph partitioning using linear and semidefinite programming, Math. Program, Ser. B 95 (2003) 91–101.
[12] R. Montemanni, L.M. Gambardella, An exact algorithm for the robust shortest path problem with interval data, Comput. Oper. Res. 31 (10) (2004) 1667–1680.
[13] A.L. Soyster, Convex programming with set-inclusive constraints and applications to inexact linear programming, Oper. Res. 21 (1973) 1154–1157.
[14] H. Wolkowicz, Q. Zhao, Semidefinite programming relaxations for the graph partitioning problem, Discrete Appl. Math. 96–97 (1996) 461–479.
[15] H. Yaman, O.E. Karasan, M.C. Pinar, The robust shortest spanning tree problem with interval data, Oper. Res. Lett. 29 (2001) 31–40.